Data visualisation and modelling.

Master's degree in Modelling for Science and Engineering

Statistical analysis of the effects of daily COVID-19 cases as a function of the mean temperature in Spain.

Author:

Joseba Hernández Bravo

Teacher:

Pere Puig

Barcelona, February 10, 2022

Contents

Abstract

In this work we will make a descriptive statistical analysis of the correlation between three meteorological variables: mean temperature, mean wind and mean daily precipitation with respect to the daily cases caused by the COVID-19 virus in the different provinces of Spain. The study has been carried out for the different waves between January 2020 and February 2022. In this study, we conclude that only for the period between 020-07-10 and 2020-11-30 shows a significant correlation of - 052 between the number of infected cases and the average daily temperature

Chapter 1

Introduction

Since the first cases of COVID-19 were detected in the Chinese city of Wuhan in December 2019, different waves and variants have spread throughout the world. To date, the virus has infected 419.693.540 people, while the number of deaths is 5.863.136. Among the most affected countries are the USA, India, England, Brazil and France. Spain, on the other hand, is the tenth most affected country according to a study based on the data collected by the government health departments.

Several studies suggest that there are significant correlation between SARS-CoV-2 virus transmission and different socio-economic and environmental factors, Kadi and Khelfaoui, 2020, Siddique et al., 2022, Livadiotis, 2020. In fact, studies such as Bashir et al., 2020 and Tosepu et al., 2020 conducted in different cities around the world such as California and Jakarta agree that the average temperature is a variable significantly correlated with the transmission of the virus.

In this work we present a statistical study of the correlation between the number of COVID-19 affected per day with respect to atmospheric conditions such as mean temperature ($^{\text{o}}$C) and mean precipitation per province (mm). The study will be carried out for the different waves of the pandemic: the first wave is established from 2020-02-15 to 2020-06-15, the second one from 2020-07-10 to 2020-11-30, the third one from 2020-12-01 to 2021-02-28, the fourth one from 2021-03-01 to 2021-06-15, and finally, the fifth one from 2021-11-01 2022-02-07.

In the first chapter we will explain the process followed from the collection of data to the pre-processing of the data for easy use in the following chapters. In the second chapter, we will give a description of the techniques used to analyse the data. In the third chapter, we will present the results obtained. Finally, in the last chapter, we will present the conclusions derived from the results obtained.

Chapter 2

Data

In this study we will use data collected from two different institutions: Red Nacional de Vigilancia Epidemiológica (RENAVE) and AEMET. It is important to note that the study will be carried out at the provincial level and that the samples will be taken on a daily basis, i.e. the data will be collected day-by-day over the period [2020-01-01, 2022-02-07].

The number of COVID-19 cases is a sum total of Number of cases with PCR laboratory test or molecular techniques, number of cases with rapid antibody test laboratory test, number of cases with antigen detection test laboratory test, number of cases with high resolution serology laboratory test and number of cases without information on the laboratory test.

Since the data has been given in different formats (.json for the meteorological data and .csv for the epidemiological one), we have made use of the python pandas library to unify them in a final table. This table, is made of 38288 rows and 8 columns:

- **provincia_iso**: defines the identification codes of the provinces[1]
- **fecha**: date in which the dates where taken.
- **num_casos**: number of cases diagnosed per day.
- **provincia**: name of the corresponding province.
- **tmed**: average daily temperature in $[°C]$.
- **wind**: average daily wind in $[m/s]$

[1] All these codes have been taken from https://es.wikipedia.org/wiki/ISO_3166-2:ES.

- **prec**: daily precipitation[2] in **[mm]** .
- **alt**: altitude of the meteorological station in **[m]**

In addition, since the study will be done in a provincial way, the general dataset has been sectioned for each of these, for its corresponding period in each wave.

All data have been stored in the following GitHub repository.

[2] All the values equal to Ip, which means that the precipitation is less than $0,1mm$, have been taken as $0mm$ precipitation.

Chapter 3

Results And Methods

In this section we will present the comparison between the daily COVID-19 cases and the rest of meteorological variables presented in chapter 2 for each autonomous community. The descriptive analysis will be examined for each of the 5 waves. These waves, corresponding to the peaks shown in the image below, are exactly bounded in the time periods: [2020-02-15,2020-06-15], [2020-07-10,2020-11-30],[2020-12-01,2021-02-28], [2021-03-01,2021-06-15], [2021-11-01 2022-02-07].

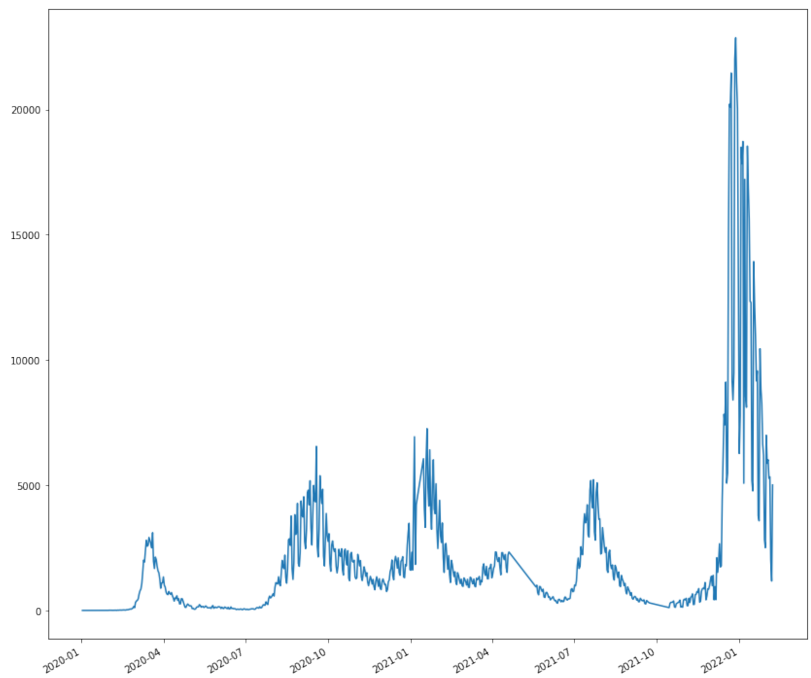

Figure 3.1: Number of daily cases of COVID-19 from January 2020 to February 2022 in Madrid.

First, In order to obtain the dependence between these variables, we have made use of Spearman's non-parametric test. As explained in the previous section, Spearman's non-parametric test does not measure the linear relationship between the variables and this makes it ideal for cases where the data are not normally distributed. This type of test is used to determine the strength and direction of the monotonic relationship between two variables. In our case, the number of cases with respect to the three meteorological variables: mean temperature l, mean precipitation, mean wind speed l, mean wind speed l, mean rainfall l and mean wind speed.

As a null hypothesis of the Spearman's test we have assumed that the pair of variables are correlated.

Looking at table 3.1 we can observe that the p-values obtained for the cases of wind and precipitation are less than 0.05. Therefore, we can not reject the null hypothesis, concluding that these variables are not correlated with the number of COVID-19 cases in each of the waves. On the contrary, for the case of the daily mean temperature, we can appreciate that for the case of the second wave, the p-value is less than 0.05. Therefore, in this case, we can affirm that the daily infections were correlated with temperature.

Given that the only statistically valid correlation found is for the first wave and for the variable "temp", we will analyse the provinces with the highest correlation in this time period.

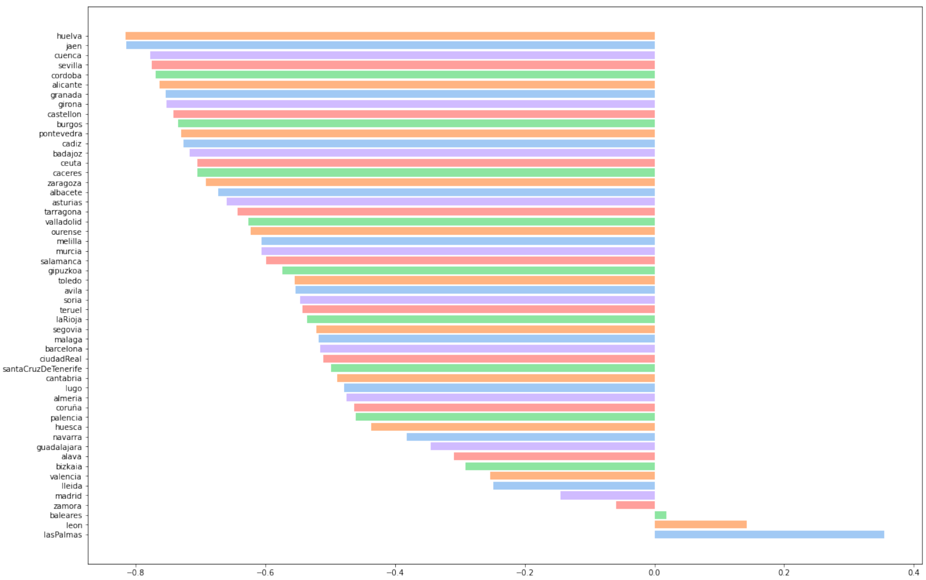

Figure 3.2: Values Spearman's correlation r per province for the second wave.

As we can see, the average of the r values gives an overall computation of -0.523, which means that these two variables are negatively related. That is, the lower the temperature, the higher the number of daily counts. The 5 provinces most correlated (negatively) were: huelva, jaen, cuenca, sevilla and cordoba.

	Temp		**Wind**		**Prec**	
Wave	r	p-value	r	p-value	r	p-value
First	−0.319(21)	0.050(21)	−0.002(15)	0.494(40)	0.163(20)	0.196(49)
Second	−0.523(34)	0.035(19)	−0.125(23)	0.291(46)	0.170(18)	0.208(40)
Third	−0.220(26)	0.187(38)	−0.056(20)	0.438(48)	0.049(18)	0.481(42)
Fourth	−0.139(48)	0.123(32)	−0.017(18)	0.423(43)	0.068(19)	0.321(27)
Fifht	−0.184(27)	0.269(40)	−0.091(20)	0.244(35)	−0.077(21)	0.294(39)

Table 3.1: Mean Spearman's correlation test values for each wave of COVID-19.

	Temp		Wind		Prec	
Province	r	p-value	r	p-value	r	p-value
Huelva	−0.816	≪0.05	−0.067	0.427	0.420	≪0.05
Jaen	−0.814	≪0.05	−0.354	≪0.05	0.314	≪0.05
Cuenca	−0.777	≪0.05	−0.349 833	≪0.05	0.230 832	≪0.05
Sevilla	−0.775	≪0.05	−0.175	0.0393	0.359	≪0.05
Cordoba	−0.769	≪0.05	−0.410	≪0.05	0.231	0.005

Table 3.2: Highest absolute Spearman correlation test values for the 2nd wave.

As can be seen in the table 3.2, although the most significantly correlated variable is temperature, the rest of the variables are correlated at the provincial level (in these cases). In fact, in the case of Jaen, we can observe that in addition to a high correlation between temperature and the case of infections, both precipitation and wind intensity show p-values lower than 0.05.

On the other hand, three provinces show a positive correlation. These three provinces are: Baleares, Leon and Las Palmas.

	Temp		Wind		Prec	
Province	r	p-value	r	p-value	r	p-value
Baleares	0.019 000	0.825	0.073 124	0.395	−0.101	0.237
Leon	0.142 321	0.309	0.382	0.004	0.192	0.168
Las Palmas	0.354 425	≪0.05	0.057	0.497	−0.044	0.597

Table 3.3: Lowest absolute Spearman's correlation test values for the 2nd wave.

Chapter 4

Conclusions

In this paper we have made a descriptive analysis of the correlation of the Covid-19 cases with the mean daily temperature, the daily wind amount and the daily precipitation amount for the five waves from January 2020 to date, February 2022.

For this we have made use of the data provided by both the meteorological agency AEMET (taken at different meteorological stations in each province) and the data collected daily by the Spanish National Epidemiological Surveillance Network (Red Nacional de Vigilancia Epidemiologia de España RENAVE).

Once all the data were collected and having changed both the nomenclature and the format of the data, we proceeded to perform a spearman test to estimate the correlation between the variable "number-chaos" and the rest of the meteorological variables. These tests have been carried out throughout the periods in which the virus has spread with greater severity. In total we have studied 5 time periods. In all the periods we observed a negative correlation of the cases with respect to the temperature and the amount of wind, while in the case of rain, we obtained positive values except for the last wave.

However, only for one of these periods we can consider the result as statistically significant. Moreover, in this particular case, the only variable with a significant correlation was temperature with a negative correlation of -0523(34). The rest of the results obtained were not statistically significant. We can therefore conclude that in the case of the second wave of COVID-19, the number of infected was higher for low temperatures than for high temperatures.

The provinces in which the correlation between temperature and the number of COVID-19 cases was found were: huelva, jaen, cuenca, sevilla and cordoba. With correlations that have reached a significant value of -0.816 in the case of huelva for example.

On the other hand, the least correlated variables, in this case (in absolute value) Baleares, Leon and Las Palmas, two of which did not show a statistically significant correlation. It should be noted that these three provinces were the only ones in which the correlation took a positive value.

Bibliography

Bashir, M. F., Ma, B., Komal, B., Bashir, M. A., Tan, D., Bashir, M., et al. (2020). Correlation between climate indicators and covid-19 pandemic in new york, usa. *Science of The Total Environment*, *728*, 138835.

Kadi, N., & Khelfaoui, M. (2020). Population density, a factor in the spread of covid-19 in algeria: Statistic study. *Bulletin of the National Research Centre*, *44* (1), 1–7.

Livadiotis, G. (2020). Statistical analysis of the impact of environmental temperature on the exponential growth rate of cases infected by covid-19. *PLoS one*, *15* (5), e0233875.

Tosepu, R., Gunawan, J., Effendy, D. S., Lestari, H., Bahar, H., Asfian, P., et al. (2020). Correlation between weather and covid-19 pandemic in jakarta, indonesia. *Science of the total environment*, *725*, 138436.

Siddique, A. B., Haynes, K. E., Kulkarni, R., & Li, M.-H. (2022). Regional poverty and infection disease: Early exploratory evidence from the covid-19 pandemic. *The Annals of Regional Science*, 1–28.

Chapter 5

Code

5.0.1 R-code

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r}
library(combinat)
library(GGally)
```

```{r}
df <- read.csv(file ='/home/joseba/Master/Master-git/DataVis
/Resampling␣Methods/proyecto/data_waves/fourth/madrid_fourth.csv')

casos <- df$num_casos
temperatura <- df$tmed
precip <- df$prec
statistical_column <- c("num_casos","tmed","wind","prec")

print (ks.test(df$num_casos, "pnorm", mean=mean(df$num_casos),
sd=sd(df$num_casos)))
print (ks.test(df$tmed, "pnorm", mean=mean(df$tmed), sd=sd(df$tmed)))
```
```

```
print (ks.test(df$wind, "pnorm", mean=mean(df$wind), sd=sd(df$wind)))
print (ks.test(df$prec, "pnorm", mean=mean(df$prec), sd=sd(df$prec)))

hist(df$num_casos)
```

Now we can calculate the spearman correlation test:

```{r}
provinces <- c('alava', 'albacete', 'alicante',
'almeria', 'asturias', 'avila',
       'badajoz', 'baleares', 'barcelona',
       'bizkaia', 'burgos', 'caceres',
       'cadiz', 'cantabria', 'castellon',
       'ceuta', 'ciudadReal', 'cordoba',
       'coru a', 'cuenca', 'gipuzkoa', 'girona',
       'granada', 'guadalajara',
       'huelva', 'huesca', 'jaen', 'laRioja',
       'lasPalmas', 'leon', 'lleida',
       'lugo', 'madrid', 'malaga', 'melilla',
       'murcia', 'navarra', 'ourense',
       'palencia', 'pontevedra', 'salamanca',
       'santaCruzDeTenerife', 'segovia',
       'sevilla', 'soria', 'tarragona', 'teruel',
       'toledo', 'valencia',
       'valladolid', 'zamora', 'zaragoza')
waves <- c("first","second","third","fourth","fifth")
path0 <- "/home/joseba/Master/Master-git/DataVis/
Resampling␣Methods/proyecto/data_waves"



provincias_vec <- c()
wave_vec <- c()

correlation_vec_temp <- c()
p_value_vec_temp <- c()
```

```
correlation_vec_wind <- c()
p_value_vec_wind <- c()

correlation_vec_prec <- c()
p_value_vec_prec <- c()

temperature_vec <- c()

for (wave in waves){
  path <- paste0(path0,"/",wave)
  for (province in provinces){
    path <- paste0(path,"/",province,"_",wave,".csv")
    df <- read.csv(file = path)

    casos <- df$num_casos
    temperatura <- df$tmed
    viento <- df$wind
    precip <- df$prec

    path <- paste0(path0,"/",wave)

    #fill vectors
    provincias_vec <- append(provincias_vec, province,
    after = length(provincias_vec))
    wave_vec<-append(wave_vec, wave, after = length(wave_vec))

    #temp
    sttrue= cor.test(casos,temperatura, method = "spearman", exact=FALSE)
    correlation_vec_temp<-append(correlation_vec_temp, sttrue[[4]],
    after = length(correlation_vec_temp))
    p_value_vec_temp<-append(p_value_vec_temp, sttrue[[3]],
    after = length(p_value_vec_temp))

    #wind
    sttrue= cor.test(casos,viento, method = "spearman", exact=FALSE)
```

```
    correlation_vec_wind<-append(correlation_vec_wind, sttrue[[4]],
    after = length(correlation_vec_wind))
    p_value_vec_wind<-append(p_value_vec_wind, sttrue[[3]],
    after = length(p_value_vec_wind))

    #prec
    sttrue= cor.test(casos,precip, method = "spearman", exact=FALSE)
    correlation_vec_prec<-append(correlation_vec_prec, sttrue[[4]],
    after = length(correlation_vec_prec))
    p_value_vec_prec<-append(p_value_vec_prec, sttrue[[3]],
    after = length(p_value_vec_prec))

    }
}

DataFrame.correlations <- data.frame(provincias_vec,wave_vec,
correlation_vec_temp,p_value_vec_temp,
correlation_vec_wind,p_value_vec_wind, correlation_vec_prec,
p_value_vec_prec)

write.csv(DataFrame.correlations,
"/home/joseba/Master/Master-git/DataVis/Resampling␣Methods/
proyecto/R/Spearman.csv",
row.names = FALSE)
‘‘‘
```

5.0.2 Pandas code

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import glob, os, json

path_covid_cases = r"/home/joseba/Master/Master-git
/DataVis/Resampling␣Methods/proyecto/casos_tecnica_provincia.csv"
path_temperatures = r"/home/joseba/Master/Master-git
```

```
/DataVis/Resampling␣Methods/proyecto/temperatura/Alava.json"
json_dir = r'/home/joseba/Master/Master-git/DataVis/
Resampling␣Methods/proyecto/temperatura'




df = pd.read_csv(path_covid_cases)
diccionario = {'C' : 'coru a', 'VI' : 'alava',
'AB':'albacete', 'A':'alicante', 'AL': 'almeria', 'O':'asturias',
'AV':'avila','BA':'badajoz',
'PM':'baleares','B':'barcelona','BU':'burgos',
'CC':'caceres','CA':'cadiz','CE':'ceuta','S':'cantabria',
'CS':'castellon','CR':'ciudadReal','CO':'cordoba',
'CU':'cuenca','GI':'girona','GR':'granada','GU':'guadalajara',
'SS':'gipuzkoa','H':'huelva','HU':'huesca','J':'jaen','LO':'laRioja',
'GC':'lasPalmas','LE':'leon','L':'lleida','LU':'lugo','M':'madrid',
'MA':'malaga','ML': 'melilla',
'MU':'murcia','NA':'navarra','NC':'navarra','OR':'ourense','P':'palencia',
'PO':'pontevedra','SA':'salamanca','TF':'santaCruzDeTenerife',
'SG':'segovia','SE':'sevilla','SO':'soria',
'T':'tarragona','TE':'teruel','TO':'toledo','V':'valencia',
'VA':'valladolid','BI':'bizkaia','ZA':'zaragoza','Z':'zamora'}

for key in diccionario:
    df.loc[df["provincia_iso"] == key, "provincia"] = diccionario[key]
df = df[df['provincia_iso'].notna()]



json_pattern = os.path.join(json_dir, '*.json')
file_list = glob.glob(json_pattern)

dfs = []
for file in file_list:
    with open(file) as f:
        json_data = pd.json_normalize(json.loads(f.read()))
```

```
        json_data['site'] = file.rsplit("/", 1)[-1]
    dfs.append(json_data)
df2 = pd.concat(dfs)

df2["provincia"] = df2["provincia"].str.lower()

df2["provincia"].replace({"las␣palmas": "lasPalmas",
"illes␣balears":"baleares", "ciudad␣real":"ciudadReal",
"sta.␣cruz␣de␣tenerife":"santaCruzDeTenerife", "la␣rioja":"laRioja",
"araba/alava":"alava", "a␣coru a":"coru a"}, inplace=True)

df2["prec"].replace({"Ip": "0,0"}, inplace=True)

to_float_array = ['tmed','velmedia','prec']
for values in to_float_array:
    df2[values] = df2[values].str.replace(',','.')
    df2[values] = pd.to_numeric(df2[values])

df2[df2["provincia"] == 'coru a']

#SOLO EJECUTAR UNA VEZ

for fech, prov , temp, viento, altitud, lluvia in zip(df2["fecha"].values,
df2["provincia"].values, df2["tmed"].values, df2["velmedia"].values,
df2["altitud"].values, df2["prec"].values):
    df.loc[(df["fecha"] == fech) &
    (df["provincia"] == prov), "tmed"] = temp
    df.loc[(df["fecha"] == fech) &
    (df["provincia"] == prov), "wind"] = viento
    df.loc[(df["fecha"] == fech) &
    (df["provincia"] == prov), "prec"] = lluvia
    df.loc[(df["fecha"] == fech) &
    (df["provincia"] == prov), "alt"] = altitud
df = df.sort_values(by=['provincia', 'fecha'])
df.to_csv("total_with_NaN.csv",index=False)
```

```
df_export = df.copy( ).reset_index()
df_export.drop(["num_casos_prueba_pcr", "num_casos_prueba_test_ac",
"num_casos_prueba_ag","num_casos_prueba_elisa",
"num_casos_prueba_desconocida", "index"], axis=1, inplace=True)

df_export = df_export.dropna()
df_export.to_csv("data_provincias/total.csv",index=False)



df_export = pd.read_csv("data_provincias/total.csv")
for provincias in df_export["provincia"].unique():
    df_provincial = df_export[df_export["provincia"] == provincias]
    df_provincial.to_csv(f"data_provincias/{provincias}.csv",index=False)

# import data for each wave
all = pd.read_csv("data_provincias/total.csv")
wave = {"first":["2020-02-15","2020-06-15"],
"second":["2020-07-10","2020-11-30"],
 "third":["2020-12-01","2021-02-28"],"fourth":["2021-03-01","2021-06-15"],
 "fifth":["2021-11-01","2022-02-07"]}

for key in wave:
    start_date = pd.to_datetime(wave[key][0])
    end_date = pd.to_datetime(wave[key][1])

    for provincias in all["provincia"].unique():
        df = pd.read_csv(f"data_provincias/{provincias}.csv")
        df['fecha'] = pd.to_datetime(df['fecha'])
        #Mask for the time period
        mask = (df['fecha'] > start_date) & (df['fecha'] <= end_date)
        df = df.loc[mask]
        df.to_csv(f"data_waves/{key}/{provincias}_{key}.csv",index=False)

# import data and select the time period

infile = "data_provincias/madrid.csv"
```

```
df = pd.read_csv(infile)

df['fecha'] = pd.to_datetime(df['fecha'])

#Mask for the time period
start_date = '2020-01-01'
end_date =  '2022-02-07'
mask = (df['fecha'] > start_date) & (df['fecha'] <= end_date)
df = df.loc[mask]
x = df['fecha']
y =  df['num_casos']

fig = plt.figure(figsize=(14,10))
fig.subplots_adjust(top=1.1, bottom=-.1) #centrar la imagen
plt.plot_date(x, y, '-')
## Rotate date labels automatically
fig.autofmt_xdate()
plt.show()

df_export = pd.read_csv("data_provincias/total.csv")
with open(f"datos_medias.csv","a", encoding="utf-8") as fout:

    for prov in df_export["provincia"].unique():
        province  = df_export.loc[df_export["provincia"] == prov]
        meant_temp = province["tmed"].mean()
        mean_cont = province["num_casos"].mean()



df = pd.read_csv("Spearman.csv")

df[df["wave_vec"] == "second"].sort_values(["correlation_vec_temp"])

second_wave = df[df["wave_vec"] ==
"second"].sort_values(["correlation_vec_temp"], ascending=False)
second_wave.reindex(second_wave["provincias_vec"])
pastel_colors = sns.color_palette('pastel')[0:5]
```

```
fig = plt.figure(figsize=(15, 10))
ax = fig.add_axes([0,0,1,1])
provinces = second_wave["provincias_vec"]
correlations = second_wave["correlation_vec_temp"]
ax.barh(provinces,correlations, color=pastel_colors)
plt.show()


Nrow = len(df[df["wave_vec"] == "first"])



print(df[df["wave_vec"] == "first"]["correlation_vec_temp"].mean())
print(df[df["wave_vec"] == "first"]["p_value_vec_temp"].mean())
print(df[df["wave_vec"] ==
"first"]["correlation_vec_temp"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"first"]["p_value_vec_temp"].std()/(Nrow-1)**(1/2))

print("-"*20)
print(df[df["wave_vec"] == "first"]["correlation_vec_wind"].mean())
print(df[df["wave_vec"] == "first"]["p_value_vec_wind"].mean())
print(df[df["wave_vec"] ==
"first"]["correlation_vec_wind"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"first"]["p_value_vec_wind"].std()/(Nrow-1)**(1/2))

print("-"*20)
print(df[df["wave_vec"] == "first"]["correlation_vec_prec"].mean())
print(df[df["wave_vec"] == "first"]["p_value_vec_prec"].mean())

print(df[df["wave_vec"] ==
```

```
"first"]["correlation_vec_prec"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"first"]["p_value_vec_prec"].std()/(Nrow-1)**(1/2))


Nrow = len(df[df["wave_vec"] == "second"])
print(df[df["wave_vec"] == "second"]["correlation_vec_temp"].mean())
print(df[df["wave_vec"] == "second"]["p_value_vec_temp"].mean())
print(df[df["wave_vec"] ==
"second"]["correlation_vec_temp"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"second"]["p_value_vec_temp"].std()/(Nrow-1)**(1/2))

print("-"*20)
print(df[df["wave_vec"] == "second"]["correlation_vec_wind"].mean())
print(df[df["wave_vec"] == "second"]["p_value_vec_wind"].mean())
print(df[df["wave_vec"] ==
"second"]["correlation_vec_wind"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"second"]["p_value_vec_wind"].std()/(Nrow-1)**(1/2))

print("-"*20)
print(df[df["wave_vec"] == "second"]["correlation_vec_prec"].mean())
print(df[df["wave_vec"] == "second"]["p_value_vec_prec"].mean())
print(df[df["wave_vec"] ==
"second"]["correlation_vec_prec"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"second"]["p_value_vec_prec"].std()/(Nrow-1)**(1/2))



Nrow = len(df[df["wave_vec"] == "third"])

print(df[df["wave_vec"] == "third"]["correlation_vec_temp"].mean())
print(df[df["wave_vec"] == "third"]["p_value_vec_temp"].mean())
print(df[df["wave_vec"] ==
```

```
"third"]["correlation_vec_temp"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"third"]["p_value_vec_temp"].std()/(Nrow-1)**(1/2))

print("-"*20)
print(df[df["wave_vec"] == "third"]["correlation_vec_wind"].mean())
print(df[df["wave_vec"] == "third"]["p_value_vec_wind"].mean())
print(df[df["wave_vec"] ==
"third"]["correlation_vec_wind"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"third"]["p_value_vec_wind"].std()/(Nrow-1)**(1/2))
print("-"*20)
print(df[df["wave_vec"] == "third"]["correlation_vec_prec"].mean())
print(df[df["wave_vec"] == "third"]["p_value_vec_prec"].mean())
print(df[df["wave_vec"] ==
"third"]["correlation_vec_prec"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"third"]["p_value_vec_prec"].std()/(Nrow-1)**(1/2))

Nrow = len(df[df["wave_vec"] == "fifth"])

print(df[df["wave_vec"] == "fifth"]["correlation_vec_temp"].mean())
print(df[df["wave_vec"] == "fifth"]["p_value_vec_temp"].mean())
print(df[df["wave_vec"] ==
"fifth"]["correlation_vec_temp"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"fifth"]["p_value_vec_temp"].std()/(Nrow-1)**(1/2))

print("-"*20)
print(df[df["wave_vec"] == "fifth"]["correlation_vec_wind"].mean())
print(df[df["wave_vec"] == "fifth"]["p_value_vec_wind"].mean())
print(df[df["wave_vec"] ==
"fifth"]["correlation_vec_wind"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"fifth"]["p_value_vec_wind"].std()/(Nrow-1)**(1/2))
```

```
print("-"*20)
print(df[df["wave_vec"] == "fifth"]["correlation_vec_prec"].mean())
print(df[df["wave_vec"] == "fifth"]["p_value_vec_prec"].mean())
print(df[df["wave_vec"] ==
"fifth"]["correlation_vec_prec"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"fifth"]["p_value_vec_prec"].std()/(Nrow-1)**(1/2))


Nrow = len(df[df["wave_vec"] == "fourth"])

print(df[df["wave_vec"] == "fourth"]["correlation_vec_temp"].mean())
print(df[df["wave_vec"] == "fourth"]["p_value_vec_temp"].mean())
print(df[df["wave_vec"] ==
"fourth"]["correlation_vec_temp"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"fourth"]["p_value_vec_temp"].std()/(Nrow-1)**(1/2))

print("-"*20)
print(df[df["wave_vec"] == "fourth"]["correlation_vec_wind"].mean())
print(df[df["wave_vec"] == "fourth"]["p_value_vec_wind"].mean())
print(df[df["wave_vec"] ==
"fourth"]["correlation_vec_wind"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"fourth"]["p_value_vec_wind"].std()/(Nrow-1)**(1/2))

print("-"*20)
print(df[df["wave_vec"] == "fourth"]["correlation_vec_prec"].mean())
print(df[df["wave_vec"] == "fourth"]["p_value_vec_prec"].mean())
print(df[df["wave_vec"] ==
"fourth"]["correlation_vec_prec"].std()/(Nrow-1)**(1/2))
print(df[df["wave_vec"] ==
"fourth"]["p_value_vec_prec"].std()/(Nrow-1)**(1/2))
```